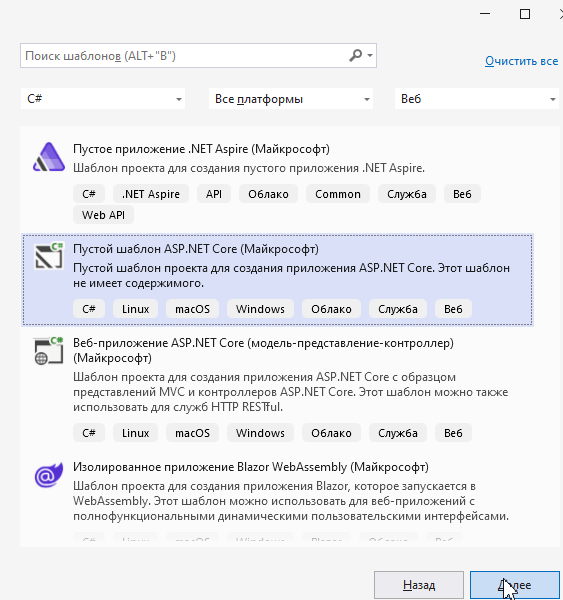
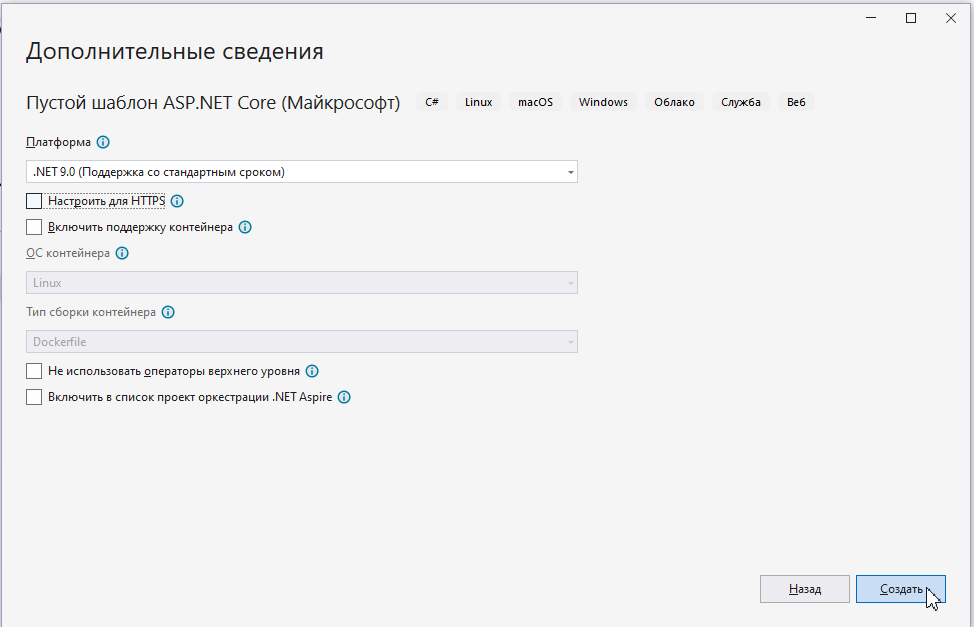
**План выполнения задания**

1. Определить сущность и её атрибуты.

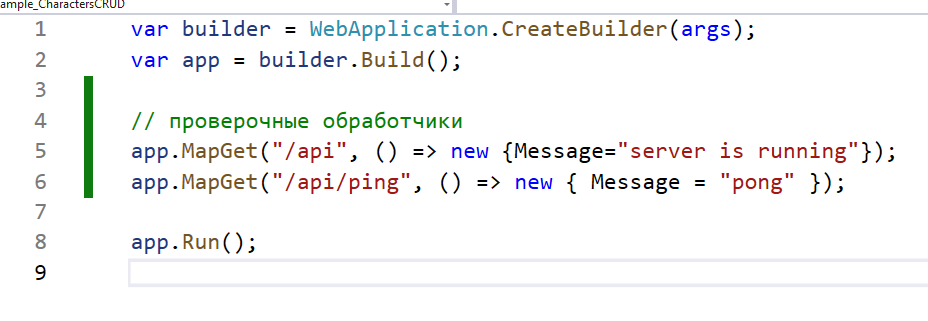
2. Создать проект пустого веб-приложения ASP.NET в Visual Studio.





Поставить самую высокую версию .NET, отключить HTTPS.

3. Добавить базовые обработчики для проверки сервера:



Настроить конфигурацию для локального запуска:



Запустить приложение и протестировать реализованные обработчики в Postman.

4. Создать класс сущности, описать её поля в виде автосвойств.

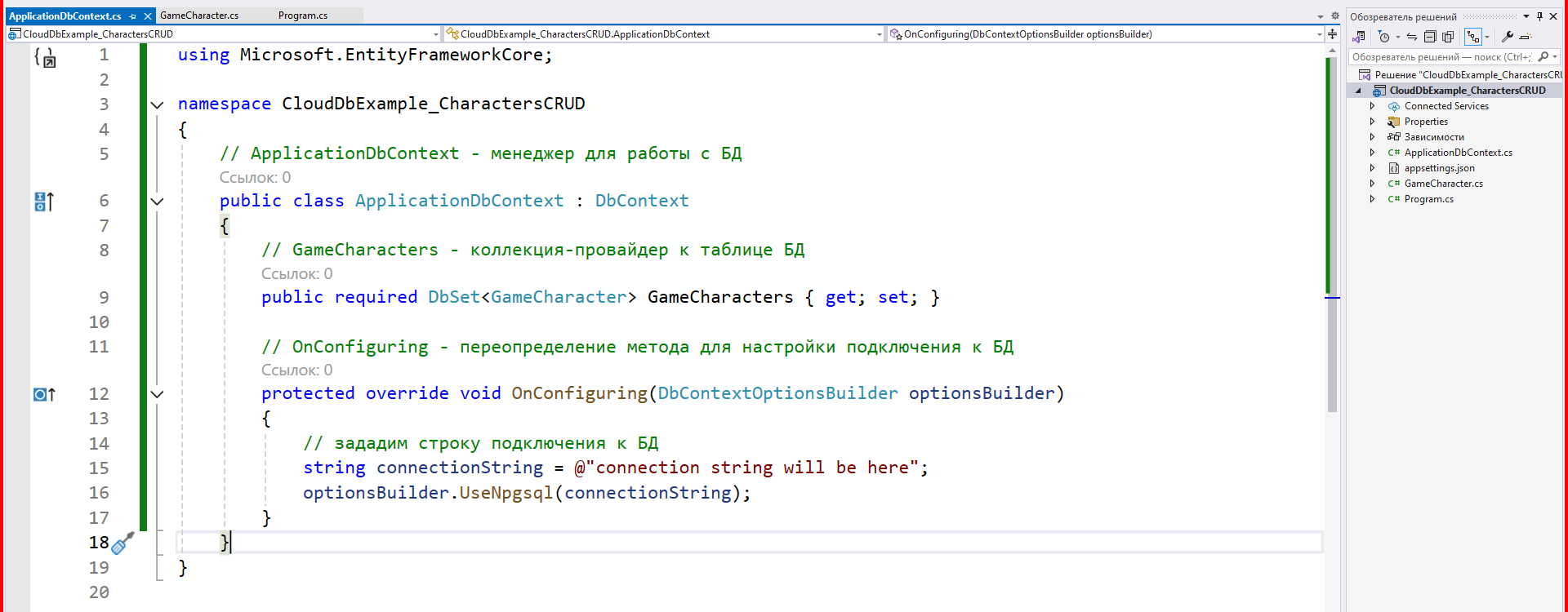
5. Установить через NuGet пакеты:

Npgsql.EntityFrameworkCore.PostgreSQL – EF для Postgres

Microsoft.EntityFrameworkCore.Tools – инструменты для работы с миграциями (в том числе)

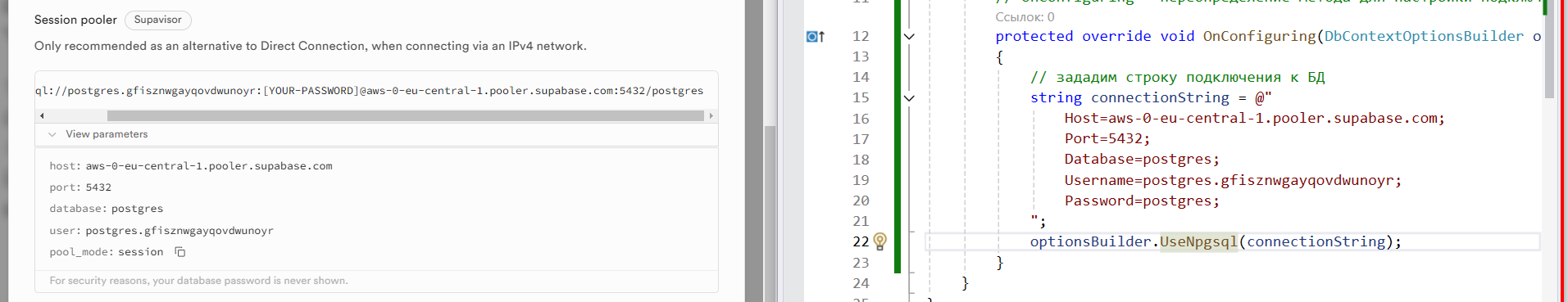
Обратите внимание, что мажорная версия устанавливаемых пакетов должна быть равна версии .NET, на которой создан проект (версии далее брать по максимуму).

6. Подготовить класс ApplicationDbContext:



7. Пора связать наше приложение и облачную БД. Для этого создать новый проект на сервисе <https://supabase.com> (либо на альтернативном, но обратите внимание, что для других СУБД понадобятся другие пакеты EF). После создания проекта подключиться к экземпляру данной БД через Table Plus – проверить доступность соединения (использовать URL-подключения через session pooler – порт 5432).

8. Задать строку подключения в классе DbContext:



9. Создать миграцию схемы и применить ее. Для этого в консоли диспетчера пакетов NuGet (только Visual Studio) выполнить команду *Add-Migration Init* для создании миграции (убедиться, что миграция создалась успешно). Далее там же применить миграцию, используя команду *Update-Database*. Убедиться в том, что миграция применилась успешно и проверить наличие созданной таблицы в БД.

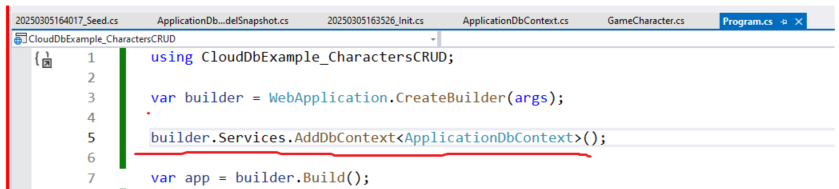
10. Создать миграцию заполнения изначальными данными. Для этого в консоли NuGet выполнить команду *Add-Migration Seed*. В новой созданной пустой миграции написать код миграции:



И применить данную миграцию, используя команду *Update-Database.*

11. Реализовать обработчики API для выполнения CRUD-операций с сущностью, попутно выполняя тестирование данных обработчиков. Реализация согласно REST.

Для каждого обработчика понадобиться экземпляр DbContext, который необходимо добавить в IoC-контейнер ASP:



После реализации каждого обработчика (или группы обработчиков) выполнять его тестирование. Пример реализации API в проекте-примере.